Improving Dutch sentiment analysis in Pattern

Lorenzo Gatti* L.GATTI@UTWENTE.NL
Judith van Stegeren* J.E.VANSTEGEREN@UTWENTE.NL

*Human Media Interaction, University of Twente, Enschede, The Netherlands

Abstract

In this paper we investigate methods for improving the sentiment analysis functionality of Pattern.nl, the Dutch submodule of Pattern, an open-source library for web mining and natural language processing. We discuss the impact on performance of three different potential improvements: extending the module's internal sentiment lexicon; removing subsets of neutral words from the sentiment lexicon; and improving the algorithm for combining multiple word-level sentiment ratings into a sentence-level sentiment rating. We evaluated the improvements on datasets from the product review domain (books, clothing and music) and a dataset of short emotional stories. The experiments show that lexicon expansion does not lead to better results; new normalization techniques, on the other hand, show a limited but consistent performance increase for sentiment ratings.

1. Introduction

Pattern[1] is an open-source Python package for NLP that is developed and maintained by the CLiPS Computational Linguistics group at Universiteit Antwerpen. The submodule for Dutch, Pattern.nl, contains a rule-based sentiment analyzer, which is based on a built-in lexicon of about 4,000 Dutch lemmata. The lexicon contains a subjectivity and polarity score for each word[2], which are used to calculate a score for an input sentence. The performance of the lexicon was evaluated in 2012 by using it to classify book reviews.

However, the applicability of Pattern in more general-domain sentiment analysis tasks is limited. For example, the sentences *Tijdens de oorlog overleed mijn jongste dochter* (During the war, my youngest daughter died) or *Ik heb het net uitgemaakt met mijn partner en nu wil ik niet langer leven* (I just broke up with my significant other and now I don't want to live anymore) will receive a neutral judgement from Pattern.nl's sentiment analysis function. Although these sentences do not explicitly contain opinions, a common focus of sentiment analysis tools, they do convey negative valence, which is not reflected in Pattern.nl's sentiment rating.

We conducted several experiments to investigate whether we could improve Pattern.nl's performance for general domain sentiment analysis tasks. Firstly, we extended the lexicon of Pattern.nl with additional Dutch words and sentiment ratings, sourced from the lexicon of Moors et al. (2013). Our hypothesis was that an increased coverage of the lexicon would lead to more non-neutral sentiment ratings for sentences outside of Pattern.nl's original domain (book reviews). We compared the effect of this extension by measuring the mean absolute error (MAE) of the original version of Pattern.nl's lexicon and our extended version against datasets of product reviews collected from the Dutch webshop Bol.com. We use the customer ratings from Bol.com as gold standard labels. Secondly, as Pattern.nl's sentiment ratings tend to be more neutral than the human ratings, we investigated whether removing neutral words from Pattern.nl's lexicon would improve its performance. Thirdly,

1. `https://www.clips.uantwerpen.be/pages/pattern-nl`
2. The word "polarity" is used in Pattern's documentation; in linguistics and psychological terms, this is normally defined as "valence", while NLP researchers might refer to this as "sentiment intensity". These all indicate the degree to which a word, sentence or paragraph is positive or negative. In this paper, these terms are used interchangeably, reflecting the usage by the sentiment analysis community.

we investigated normalization methods for Pattern.nl's sentiment rating algorithm, i.e. alternative ways of combining sentiment ratings of words to form the sentiment rating for complete sentences. We evaluated the normalization improvement on various datasets: book reviews (the original domain of Pattern.nl), product reviews of clothing and music albums, and emotional stories.

The rest of the paper is structured as follows: in Section 2 we describe other lexicon-based and rule-based systems for sentiment analysis, how large their lexicons are and how they were created. In Section 3, we describe in detail how the sentiment analyzer of Pattern works, its lexicon, and some of its limitations. Section 4 describes the datasets we used in our experiments. Section 5 reports multiple experiments aimed at circumventing these limits. Finally, in Section 7 we draw our conclusions.

In the rest of this paper, we will say 'Pattern' instead of 'the sentiment analysis functionality of Pattern.nl' for brevity.

2. Related work

Sentiment analysis is a traditional NLP task, similar to tasks such as part-of-speech tagging and parsing. During the early days of this subfield, a large number of systems were mostly based on lexicons such as SentiWordNet (Baccianella et al. 2010), sometimes adding rules to combine individual word scores, such as in the work of Neviarouskaya et al. (2011). While most state-of-the-art systems are supervised and based on deep learning techniques (Zhang et al. 2018), these usually require users to train them on specific annotated corpora. This can be challenging due to a lack of language resources, technically proficient users or available computational resources. For these reasons, few lexicon-based and rule-based systems are still actively developed and used for research; in this section, we review some of these, with particular attention to the ones that are available for Dutch.

LIWC (Linguistic Inquiry and Word Count) (Pennebaker and Francis 2001) is the oldest among these tools. Its goal is broader than simple sentiment analysis, in that it can be used for detecting a number of different aspects, such as emotional tone, pronouns usage, senses-related terms and swearwords. Its *PosEmo* and *NegEmo* output categories are however commonly used in a way comparable to a simple sentiment analysis tool. As the name suggests, LIWC is based on counting words. More precisely, it simply reports the proportion of words in a text that belong to one of its categories, such as the aforementioned *PosEmo*. The Dutch translation (Boot et al. 2017) contains 6,614 words, however only 2,018 words belong to the *PosEmo* or *NegEmo* category. The tool is often a popular choice for psychologists and social scientists studying corpora, due to its validated lexicon and an easy-to-use graphical interface. It is, however, a commercial closed-source product. No rules for detecting negation or intensifiers are implemented.

One of the most used rule-based systems is Vader (Hutto and Gilbert 2014), possibly due to its inclusion in NLTK, the popular Python library for text processing. Vader is specifically targeted to "microblog-like" contexts; its lexicon is an expansion of multiple lexicons, i.e. ANEW (Bradley and Lang 1999), the General Inquirer (Stone et al. 1966) and LIWC (Pennebaker and Francis 2001), plus "numerous lexical features common to sentiment expression in microblogs, including [...] emoticons, sentiment-related acronyms [...], and commonly used slang" (Hutto and Gilbert 2014). These 9,000 terms were then annotated via crowdsourcing, and multiple aggregation heuristics were manually derived from a collection of positive and negative tweets. Vader returns a score between -1.0, indicating strong negativity, and 1.0, to indicate strong positivity. Vader is open source, its lexicon and rules easy to inspect, and thus its output can be explained very easily; however, it is only available for English.

SentiStrength (Thelwall et al. 2012) was created in a similar way. It extends the General Inquirer and LIWC lexicons with ad-hoc words derived from web corpora. The words were manually annotated and their weights were refined by a supervised ML algorithm. SentiStrength also uses combination rules to increase or decrease the score of certain word sequences. However, it differs from Vader in that it always returns two scores, one in the $[1; 5]$ range and one in the $[-1; -5]$ range. The former

indicates the "positivity level" of the sentence/text, while the latter indicates its negativity level. It can, however, also report scores in more easily interpretable formats (presumably through an internal combination of these two ranges), such as binary or ternary labels (positive/negative/neutral) or a single scale from -4 to +4. While originally developed for English, a Dutch version is available online[3], although no details on its dictionary[4] or rules are available, and the code is not open source.

Hogenboom et al. (2014) explored extending a rule-based sentiment analyzer using lexical resources. They created a new sentiment lexicon for Dutch, based on a small set of seed words and a semantic graph with word relations. Although the goal of the authors was to extend a sentiment analysis tool to support different languages, i.e. adapt an English tool to the Dutch language, their method could also be used to extend a Dutch system with additional Dutch lexical resources.

The focus of this work is a different lexicon-based system: the sentiment analysis functionality of Pattern.nl. The following section will detail its lexicon and composition rules.

3. Pattern

Pattern[5] is an open-source Python package for "web mining" that is developed and maintained by the CLiPS Computational Linguistics group at Universiteit Antwerpen (De Smedt and Daelemans 2012a). In addition to web scraping and machine learning functionality, it provides a comprehensive set of natural language processing tools, such as part-of-speech taggers, n-gram models, and a WordNet interface, for multiple languages. It supports English, Spanish, German, French, Italian and Dutch.

3.1 Sentiment analysis in Pattern.nl

A noteworthy feature of Pattern.nl, the submodule for Dutch, is a rule-based sentiment analyzer, which is based on a built-in lexicon of 3,304 unique Dutch lemmata (De Smedt and Daelemans 2012b), 97% of them being adjectives. Each entry of the lexicon includes a word lemma, the word polarity, and other information such as a subjectivity score and a definition. Multiple entries for the same words are permitted, as they indicate multiple senses of the same lemma with different polarities. An excerpt from the lexicon of Pattern.nl (in XML format) is shown in Listing 1. Here, it can be seen how *goed* (good, well) appears twice, once with a quite positive polarity score of 0.9, and once with a more moderate score of 0.5. Polarity can range from -1 to 1, with the two extremes indicating, respectively, strong negativity and strong positivity. The third word, *verschrikkelijk* (terrible), is very negative, with a polarity score of -0.9.

A high-level overview of the algorithm that Pattern.nl uses for sentiment analysis is the following:

1. find every chunk inside a sentence;
2. calculate the score for each chunk using rules;
3. average the score of chunks to compute the final score for that sentence.

Chunks are groups of words that are combined on the basis of the sequence of part-of-speech tags. They are analogous to constituents in constituency parsing, with the exception that they are not recursive but "flat". Pattern distinguishes between nominal, verbal, adverbial, adjectival and prepositional chunks. To compute the score of a chunk, Pattern averages the polarity score of all constituent words for which it has an entry in the lexicon.

There are a few exception to this rule. For words that have multiple entries in the lexicon, such as *goed*, Pattern uses the average sentiment score of all entries. Thus, like most rule-based sentiment analysis systems, it lacks a word sense disambiguation component. The ambiguity problem can be partially addressed by considering the part of speech (POS) tags. If given a POS-tagged input text,

3. `http://sentistrength.wlv.ac.uk/#Non-English`
4. The English version recognizes 2,546 word stems.
5. `https://github.com/clips/pattern`

```
<word form="goed" cornetto_id="r_a-11143" cornetto_synset_id="c_168"
    wordnet_id="a-01123148" pos="JJ" sense="in orde" polarity="0.6"
    subjectivity="0.9" intensity="1.0" confidence="1.0" />

<word form="goed" cornetto_id="r_a-11144" cornetto_synset_id="c_680"
    wordnet_id="" pos="JJ" sense="correct" polarity="0.5" subjectivity="0.9"
    intensity="1.0" confidence="1.0" />

<word form="verschrikkelijk" cornetto_id="r_a-16011" cornetto_synset_id="
    c_267" wordnet_id="a-01385255" pos="JJ" sense="heel akelig" polarity="
    -0.9" subjectivity="1.0" intensity="1.9" confidence="1.0" />
```

Listing 1: An excerpt of the lexicon of Pattern.nl

Pattern only averages the relevant entries from its sentiment lexicon. For example, for the input text *Het is goed* (It is good), it will average all *goed* senses marked with a adjective tag in its sentiment lexicon, and ignore all *goed* senses that are tagged as a noun.

Pattern also distinguishes between normal sentiment words and modifiers using part of speech tags. In Dutch, some words (like *verschrikkelijk*) can be used as both an adjective or as adverb, e.g. *verschrikkelijk mooi* (lit. terribly beautiful). If *verschrikkelijk* occurs as adjective in a sentence, Pattern will take the polarity score from the lexicon into account when computing the score for the complete sentence. However, if *verschrikkelijk* is used as an adverb, Pattern will label it as a intensifier and instead modify the polarity score for the adjective that follows. In the case of *verschrikkelijk mooi*, the polarity of *mooi* will be multiplied by 1.9, i.e. the intensity score shown in Listing 1. Finally, additional rules deal with negations (invert the polarity of what follows), emoticons and exclamation marks.

3.2 Usability

To the best of our knowledge, Pattern.nl is the only publicly available system for sentiment analysis for Dutch that is open-source, free,"plug and play" and easy to use. While other systems are described in the literature (Van Atteveldt et al. 2008, Schrauwen 2010), and various Dutch BERT implementations are available (Vries et al. 2019, Delobelle et al. 2020), they usually require either a re-implementation of what is documented in the paper, including sourcing appropriate training data, or a degree of technical knowledge that makes it less accessible to scholars from a non-technical background.

While this is not necessarily a problem for NLP practitioners, sentiment analysis tools are also used by researchers in the social sciences and humanities. Compared to other sentiment analysis tools for Dutch, the entry barrier for Pattern.nl is much lower. Pattern is supported by `pip`, the package manager for Python, which makes installation easy for potential users. After installing the library, the code required to get the prediction for a sentence is two lines long:

```
> from pattern.nl import sentiment
> print(sentiment('Een onwijs spannend goed boek!'))
(0.6875, 0.90)
```

The two returned numbers are the polarity and subjectivity scores for the input sentence.

Furthermore, Pattern's sentiment analysis is transparent. With the following code, we can check why Pattern assigns a certain score to the sentence *Een spannend boek dat ik echt goed vind* (A thrilling book that I really like):

```
> sentiment('Een spannend boek dat ik echt goed vind!')
(0.525, 0.9)
```

```
> sentiment('Een spannend boek dat ik echt goed vind!').assessments
[(['spannend'], 0.05, 0.8, None), (['echt', 'goed', '!'], 1.0, 1.0, None)]
```

By inspecting the `assessments` field, we can easily check which words are recognized and/or have a score in the lexicon. The word *boek* does not have a sentiment score, while *spannend* (thrilling, exciting), *echt* (really) and *goed* do. The exclamation mark acts as an intensifier. We can also inspect the score assigned to each chunk, namely 0.05 for *spannend* and 1.0 for *echt goed*. This easily explains the final sentence score, which is the average of all chunks in a sentence.

3.3 Limitations

When we tested Pattern.nl's sentiment analyzer on a collection of Dutch emotional stories, we noticed that various valence-laden sentences received a neutral rating despite their emotional content. By inspecting Pattern's sentiment assessment for those sentences, we found that Pattern often does not take valence-loaded words, such as *slechte* (bad) or *vreselijk* (horrible), into account during the computation of the sentiment rating. Our tests suggested that Pattern is biased towards words that occur in product reviews. Given that its lexicon was developed and evaluated on a dataset of book reviews, this is not surprising. However, this specificity severely limits the applicability of Pattern's sentiment analyzer to texts from different domains.

4. Data

In Section 5, we will describe various modifications to Pattern and our experiments to evaluate these modifications. To test whether the modifications actually improve the performance, we need a dataset with annotated sentiment values. Unfortunately, as it is the case for many under-resourced languages, the choice of datasets with sentiment annotations is quite limited for Dutch. The original authors of Pattern tested the performance on a book reviews corpus that they collected themselves (De Smedt and Daelemans 2012b), for which they reporting an accuracy of more than 80%. However, the original evaluation corpus is not publicly available.

Other Dutch evaluation corpora, such as the dataset developed for SemEval 2016 (Pontiki et al. 2016), or the recent dataset by Van der Burgh and Verberne (2019), include only binary (positive, negative) or ternary (positive, negative, neutral) sentiment annotations. As Pattern's sentiment ratings are continuous on a scale of -1 to 1, these corpora seem limiting for measuring the performance of Pattern.

4.1 Reviews from Bol.com: books, clothing and music

In order to evaluate Pattern's extended lexicon on a multiclass dataset, we scraped more than 60,000 book reviews from Bol.com. Each review consists of a title, a text, and a rating score between 1 and 5 stars. For our experiments, only the full text and the ratings are considered. The review datasets contained a high class-imbalance, with an overwhelming bias towards the positive class: 37,589 books received 5 stars, while less than 5,000 items in total were rated with 1 or 2 stars. To prevent this imbalance from skewing the results, we subsampled each class to the size of the least represented class. Some statistics for the resulting dataset are reported in Table 1.

As for measuring the performance, research in sentiment analysis often reports accuracy or F1-score; this would be possible in our case as well, but measuring these would require using a threshold to determine when the numeric output by Pattern corresponds to a "true positive". Not only this, but any mistake given by the system would be weighted the same, while – intuitively – misclassifying a "very positive" review as "very negative" is much worse than misclassifying it as "positive". Thus, for our experiments we report the Mean Absolute Error (MAE).

Dataset	Size	Original annotation	Balanced	Average length
Book reviews	10,930	5 classes	yes	80 tokens
Clothes review	2,420	5 classes	yes	20 tokens
Music albums reviews	5,396	5 classes	yes	41 tokens
Emotional stories	120	9 classes	no	149 tokens

Table 1: Statistics about all the datasets used

As mentioned in Section 3, the output of Pattern's sentiment analysis module is a number between -1 and 1. Our dataset, however, a scale of 1 to 5 "stars". To map the book review star ratings to Pattern's sentiment rating range of $[-1, 1]$, we used min-max normalization.

An important requirement for testing the performance of Pattern on domains other than book reviews (per Hypothesis IV in Section 5.3) is at least one dataset with fine-grained annotations for a different domain. To this end, we scraped several thousand reviews for music albums and clothing items from Bol.com. The data was normalized according to the same method as we used for the book reviews.

Our datasets with reviews for books, music albums and clothing items are available on Github.[6]

4.2 Emotional stories

The datasets described above are still in the general domain of reviews, and quite different from our initial "personal memories". To address this problem, we sourced a fourth labelled corpus from the Tilburg center for Cognition and Communication of the University of Tilburg (Braun et al. 2020).

This corpus contains 120 "emotional stories": participants to an experiment were asked to write about three memories from their life, and indicate the valence of the story with a number between 1 and 9. The average length of stories is 149 words, while reviews are usually much shorter (73, 17, 33 words on average for book, clothes and music reviews respectively). This allows us to test Pattern and its potential improvements on a domain relevant to social science research. Furthermore, the emotional stories dataset has 9 gold standard labels instead of 5. Since the rating scale is more fine-grained, we should be able to better see differences in performance.

The corpus of emotional stories is imbalanced, as the valence distribution follows an inverted bell curve, with most memories being very positive or very negative. Due to the small dataset size, however, we decided not to perform any class normalization. Also in this case, min-max normalization was performed on the gold labels to bring them in the $[-1; 1]$ range. The statistics for all dataset are shown in Table 1. The data is available on OSF.[7]

5. Experiments

Our goal was to improve Pattern, so that it can perform better on multiple domains, different from book reviews. We investigated various improvements, both for the internal lexicon and the algorithm for computing sentiment ratings. In this section, we describe our experiments towards this goal and discuss the results.

6. https://github.com/hmi-utwente/pattern-nl-sentiment-improvement
7. https://osf.io/ekqmj/

Dataset	Original	Extended
Coverage	15.29%	29.83%
Mean Average Error	**0.525**	0.567

Table 2: Coverage and MAE for original and extended lexicon on book reviews

5.1 Lexicon expansion and neutral word removal

5.1.1 Hypothesis I

Our first hypothesis is that *the performance of Pattern can be improved by expanding the lexicon, so that the sentiment lexicon covers valence-laden words, such as "zelfmoord", that are currently missing in the internal lexicon.*

5.1.2 Method

To extend Pattern's ability to detect polarity-laden words, we needed another lexicon with fine-grained valence annotations. To the best of our knowledge the largest resource for Dutch with fine-grained annotation is the affective norms collected by Moors et al. (2013). This resource[8] contains ratings for 4,300 Dutch words, all with ratings of valence, arousal, dominance, and age of acquisition. Words in Moors et al.'s lexicon are not sense-disambiguated.

The valence ratings were collected using a 7-point Likert scale, where a value of 7 indicates a strong positive sentiment and 1 a strong negative sentiment. We rescaled this value into the $[-1; 1]$ range, using min-max normalization, to make it consistent with the range used by the Pattern lexicon.

To create the extended lexicon, we removed stop words from Moors' lexicon using NLTK's Dutch stop word list[9] and inserted the remaining lemmata in Pattern's lexicon file. The new lexicon consists of 8,217 entries (7,603 unique entries, considering the sense-disambiguated words of the original lexicon). 726 entries are shared between the two resources. The correlation between their polarity values is 0.84, which suggests Moors' lexicon is a suitable extension for Pattern's sentiment lexicon.

5.1.3 Results and Discussion

We calculated sentiment ratings for the reviews in our book review dataset, and computed the Mean Average Error between our gold standard ratings and the rating computed by the original and extended lexicons. The results, shown in Table 2, confirmed that the larger lexicon almost doubles the coverage, i.e. the number of tokens recognized by Pattern out of the total number of tokens in a sentence. However, the performance of the extended lexicon was actually lower. We plotted a random subsample of 100 reviews in Figure 1, along with the gold score to predict, the predicted score of the original lexicon and the one of the extended lexicon. From this graph, we can see that the extended scores are usually closer to 0 compared to the original scores. We suspected this was due to the high number of neutral words in Moors: since Pattern, at its core, is mostly computing averages of word scores, adding a large number of neutral words to its lexicon would lead to more neutral sentiment ratings. Looking at the distribution of the 4,300 lemmata in Moors' lexicon suggested this might indeed be the case (Figure 2).

8. `http://crr.ugent.be/archives/878`
9. In a normative lexicon like Moors', stop words they can have a sentiment rating. For example, the word *niet* (not) has a slightly negative polarity. While this makes sense in the context of psycholinguistics, it is less useful in sentiment analysis, where -intuitively- they are used as function words, and their polarity in isolation is not relevant for the task.

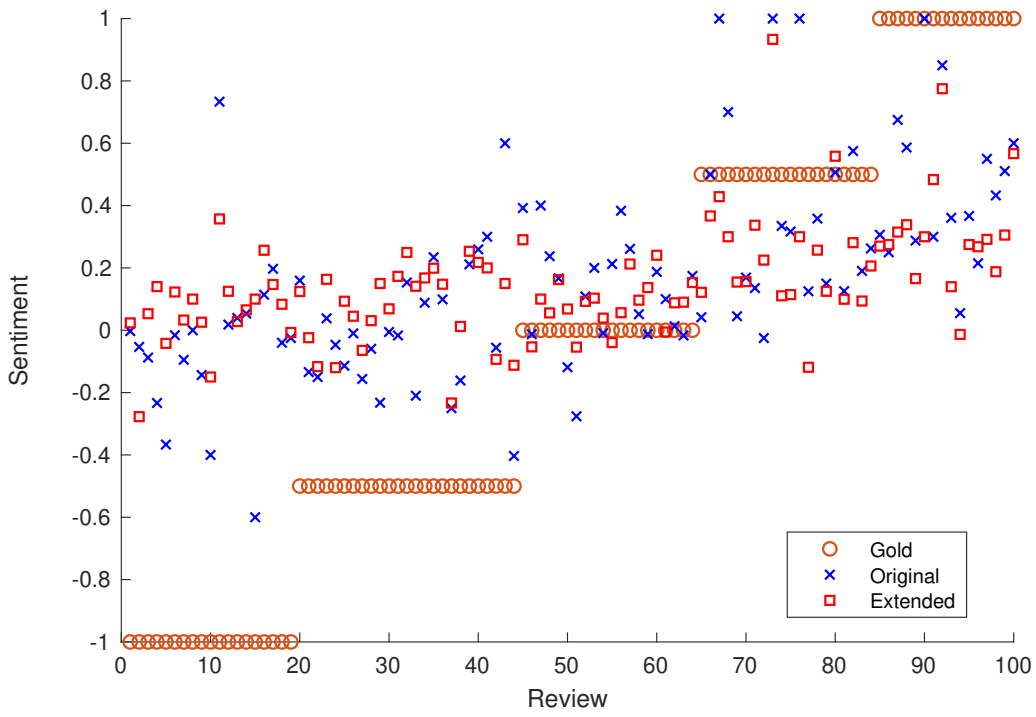

Figure 1: Predicted versus actual sentiment ratings on a subsample of book reviews

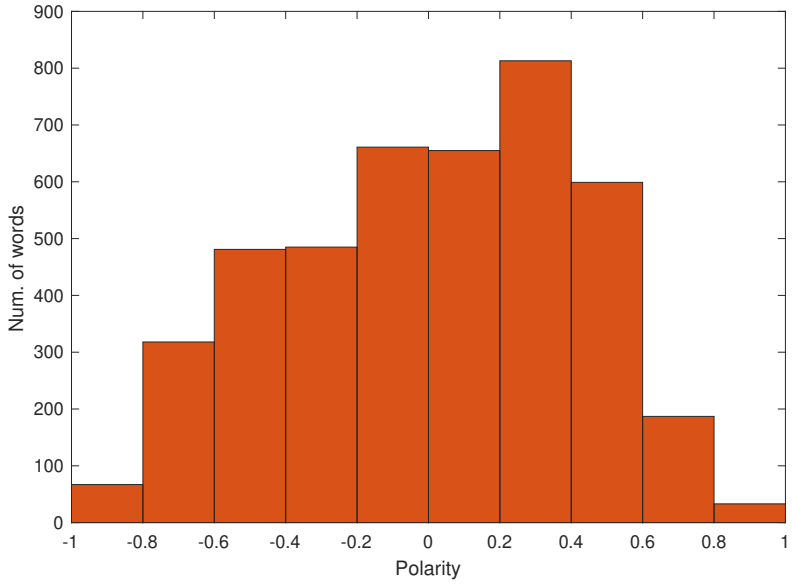

Figure 2: Distribution of word ratings in Moors' lexicon

5.1.4 Hypothesis II

This led us to the second hypothesis: *expanding the Pattern lexicon will result in a higher performance; however, neutral words have to be removed first from the lexicon of Moors et al., or they will flatten the predictions towards 0.*

5.1.5 Method

To test our hypothesis, we iteratively removed bins of words from Moors' lexicon, with bins centering around a valence of 0, with increasingly large thresholds. The reduced lexicon was then used to expand Pattern's lexicon. We started removing words with a valence rating in the $[-0.05; 0.05]$ range, and increased the size by 0.1, thus removing words in the $[-0.15; 0.15]$, $[-0.25; 0.25]$ ranges, and so on, until we reached the $|0.95|$ threshold. The rationale behind this is that it is hard to determine what constitutes a neutral word. Due to the way the algorithm works it might be beneficial to include only very loaded words.

We then tested all these lexicons, using the same evaluation method and performance measure as we used for Hypothesis I.

5.1.6 Results and Discussion

The results for this experiment are shown in Figure 3. The blue horizontal line represents the performance of the original Pattern lexicon, while the red squares are the various "threshold lexicons". As the graph shows, each successive removed bin increases the performance, compared to the full extended lexicon. However, the only actual improvement (MAE = 0.524) compared to the original lexicon (MAE = 0.525) is achieved when all the words falling between the $[-0.95; 0.95]$ range are discarded. Not only the magnitude of the improvement is not significant, but of the initial 4,300 words of Moors' lexicon, only 6 were included in the extended lexicon in this setting. Thus, not only the performance is essentially unchanged, but so is the coverage, rendering the expansion useless.

5.2 Normalization formulae

While the experiments of the last section showed no improvement, we considered the possibility that the limiting factor might be the way Pattern combines word scores. As we have mentioned in the previous sections, Pattern averages chunk scores together to obtain a final score for the whole text. This kind of normalization technique guarantees that the final prediction is in the range $[-1; 1]$, and ensures easy interpretability of the output. However, averaging scores is problematic especially for the most extreme labels of our dataset: to obtain a prediction of 1, every chunk in the sentence must have a score of 1, which in turn implies that almost every word[10] must have a score of 1. The more words are recognized by Pattern's lexicon, the less likely it is for a sentence to receive an extremely positive or negative sentiment rating.

5.2.1 Hypothesis III

These considerations led us to the next hypothesis: *the benefits of a larger lexicon are hindered by the formula used for normalization (i.e., the average). Different formulae, less influenced by the presence of neutral values, will perform better.*

5.2.2 Method

This hypothesis was tested by modifying the part of Pattern's code that computes the final score from all the chunks, while using the same lexicon described in Section 5.1.2, i.e. the full extended lexicon. A potential replacement candidate for the original "mean" function must take as input the

10. This is of course an extreme simplification, as intensifiers and negations can play a role as well.

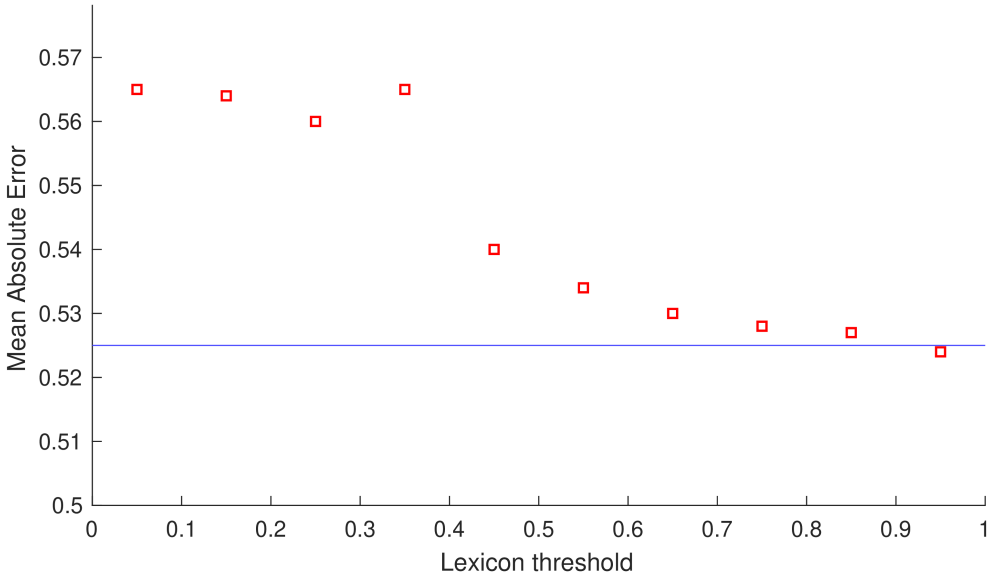

Figure 3: Performance comparison for the various threshold lexicons, represented by red squares. The blue line represents the performance of the original Pattern lexicon. A lower MAE means better performance.

individual chunk scores and return a number in the $[-1; 1]$ range. In our case, we chose to sum all chunks scores, and use this as input for two functions: the hyperbolic tangent (tanh), and the same normalization function used by the Vader tool for sentiment analysis (Hutto and Gilbert 2014). Vader's normalization function is

$$SentenceScore = \frac{\sum_{chunks}}{\sqrt{(\sum_{chunks})^2 + \alpha}}$$

where $\sum_{chunks}$ is the sum of the chunks scores, and α is a normalization factor.

Both the hyperbolic tangent and Vader's normalization function are monotonically increasing; the idea behind them is that every chunk score is a piece of evidence in favor of a positive or negative verdict. How quick the absolute certainty is reached (i.e., an extreme rating of -1 or 1) depends on the function. In Figure 4, a plot of each function is shown.

For Vader's normalization function we show the effect of different values of the hyperparameter α. The optimal value for α is dependent on both the average length of the data to annotate, and the lexicon annotations. Short texts requiring a smaller α, or they cannot converge to the maximum value even when consisting of very positive words only. In the code by Hutto and Gilbert, α is set to 15. However, in the case of Vader, word sentiment values are in the $[-3.9, 3.4]$ range, instead of the $[-1, 1]$ range of Pattern's lexicon, which explains why we observed better performance with different values for α.

We compared both the regular and extended lexicons with the new normalization methods, and tried different values of α. We report the results with α set to $\{0.25, 0.5, 1\}$. We also tried smaller (0.1) and higher values, up to the original value of 15, but the MAE in those cases is always higher than in the results here reported.

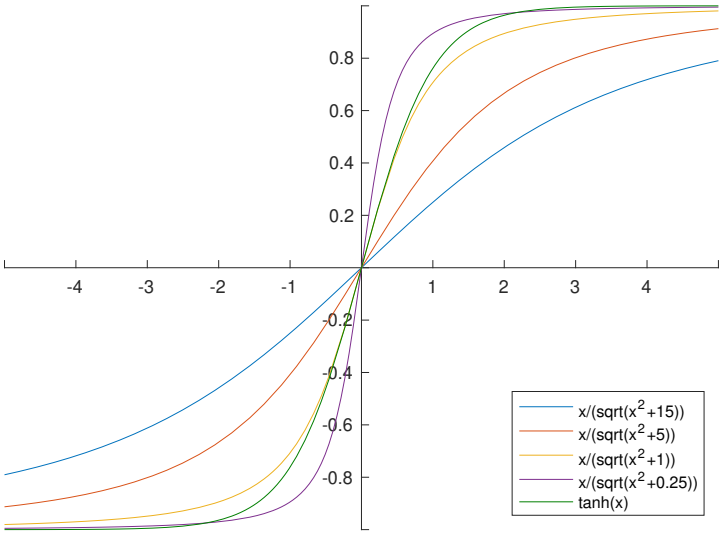

Figure 4: Different normalization functions. The green line represents the hyperbolic tangent (tanh) function. The other four lines represent Vader's normalization function with different values for hyperparameter α: 0.25, 0.5, 1 and 15.

Normalization function	Original lexicon	Extended lexicon
Mean	0.525	0.567
Hyperbolic tangent (*tanh*)	0.526	0.575
Vader, $\alpha = 0.25$	**0.520**	0.586
Vader, $\alpha = 0.5$	**0.520**	0.575
Vader, $\alpha = 1$	0.527	0.576

Table 3: Mean Average Error (MAE) on book reviews for original and extended Pattern lexicons using different normalization functions. Lower MAE means better performance. The original Pattern uses a mean function, so we use that as our baseline. Vader's normalization function improves the performance on book reviews for Pattern's original lexicon, if we set α to 0.25 or 0.5.

5.2.3 Results

The results of this experiment are shown in Table 3. Using Vader's normalization function does improve the Mean Absolute Error on books review, when the hyperparameter α is set to 0.25 or 0.5. This happens only when the normalization is used in conjunction with the regular lexicon, and the improvement is in any case marginal.

5.3 Additional domains

The results in the previous section ruled out any positive effect given by the lexicon extension, and seemed to suggest a minor positive benefit from using a different normalization function. However, some potential issues came to mind. The first concerns the performance gain of the new normalization formula. Based on the performance on a modified Pattern on book reviews, we cannot determine whether Vader's normalization is a consistent albeit marginal improvement or we are just overfitting to this particular dataset. On top of this, our starting point in Section 3.3 was the analysis of personal memories. We were, up to now, testing potential improvements on book reviews, i.e. a corpus that is very different from our target domain. Book reviews are hardly representative of generic text[11], and the original lexicon of Pattern has been created starting from adjectives extracted by book reviews from Bol.com. It might very well be that the extended lexicon is not showing any improvement due to Pattern having a large coverage of words in this domain[12] while lacking in other domains, as our initial tests suggested. This led us to formulate our last hypothesis.

5.3.1 Hypothesis IV

The lexicon extension is beneficial only for domains outside book reviews; the wider the gap between those domains and book reviews, the more the lexicon extension will increase the performance. A similar effect can possibly be seen with the normalization techniques.

5.3.2 Methods

To test our hypothesis, we applied Pattern to three other datasets, namely the clothing and music reviews and emotional stories, as described in Section 4. We compared the performance for both the original and extended lexicons, and a variety of normalization functions.

5.3.3 Results and Discussion

The results for all the datasets we collected are shown in Table 5. Results for using the hyperbolic tangent as normalization function were omitted for brevity, as they are in every case worse than those obtained with Vader normalization. As can be seen, also in this case the expanded lexicon obtained consistently larger error, disproving once more the assumption that a larger lexicon would be useful at least in different domains. The performance penalty is actually the largest among all datasets (from 0.656 to 0.708 for chunk averaging, i.e. 0.05 points worse), suggesting that either increasing the lexicon is outright a bad idea, or (perhaps more likely) that Moors et al.'s lexicon is simply not the right resource for this task. The latter option is perhaps supported by the fact that regular Pattern's coverage for the emotional stories is much lower compared to the review datasets, as can see from Table 4, but the performance on reviews is consistently higher[13]. Thus, it might still be the case that, for emotional stories, Pattern's performance is greatly impacted by a limited lexicon – however extending it with Moors et al. (2013) is not the solution.

As for different normalization techniques, their effect seems mostly positive, and in the case of emotional stories the improvement is almost 8%. Figure 5 shows that the effect of Vader's normalization is, as expected, to "boost" the score of Pattern's output by pushing the ratings towards the extremes: positive predictions are more positive, negative predictions are more negative. In the central part of the graph, where neutral texts are located, the difference between the average and Vader normalization is less pronounced. It is also worth noting, however, that in many emotional

11. Furthermore, book reviews might describe items of the plot and characters of the book itself, biasing a sentiment analyzer that cannot identify and distinguish different aspects, as in the case of Pattern.
12. De Smedt and Daelemans (2012b) write that "precision and recall do not increase by adding more adjectives [because] 90% of top frequent adjectives is already covered in [the seed lexicon used to bootstrap Pattern], adding more words has a minimal coverage effect".
13. Although it is worth noting that in this experiment it is impossible to separate the effects of domain, average text length, and difference in rating scales.

Dataset	Regular	Extended
Book reviews	15.29%	29.83%
Clothes review	21.20%	34.53%
Music albums reviews	16.54%	20.20%
Emotional stories	9.34%	27.43%

Table 4: Coverage for regular and extended Pattern on all datasets

stories the regular lexicon is predicting a negative result for a positive sentence, and vice versa. In those cases, a different normalization technique cannot improve the results (and will actually worsen them), but only a better lexicon and composition rules can potentially make a difference.

In any case, the best value for the α parameter is highly dataset-dependent, and its intuition (i.e., datasets with shorter sentences should perform better with smaller value of α) is disproved by the experiments. For emotional stories (avg. 149 tokens/story) the best value is 0.25, while Music album reviews (avg. 41 tokens/review) perform best with $\alpha = 1$. Given these results, we cannot indicate a reasonable "one size fits all" value for this hyperparameter; our suggestion is to annotate a subset of the data to process with Pattern.nl, and tune the value of α on those.

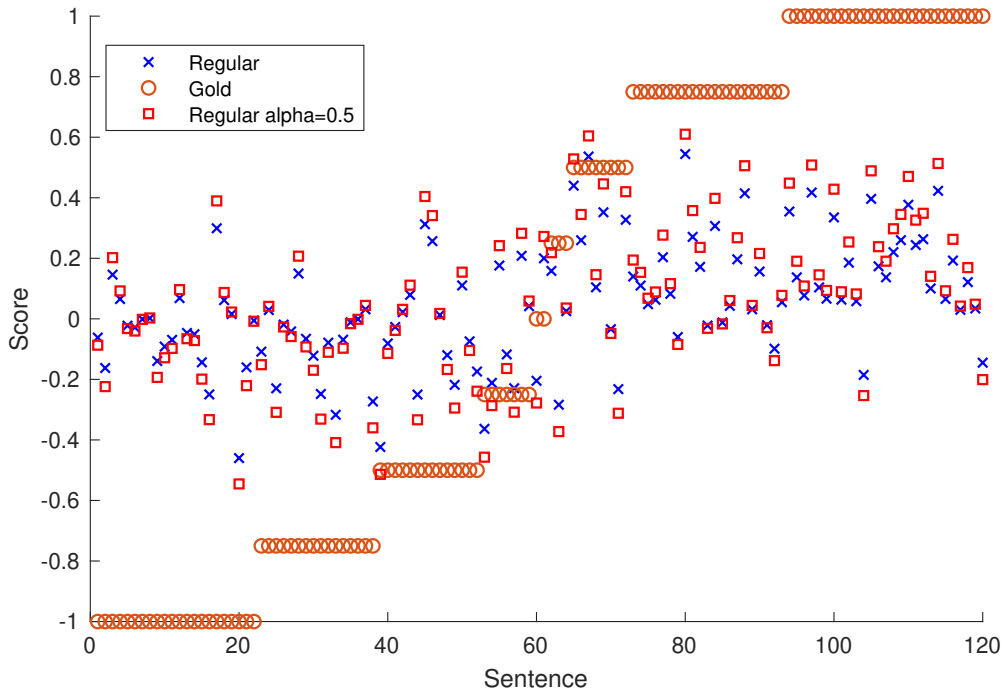

Figure 5: Comparison on emotional stories between regular Pattern with and without Vader normalization

Dataset	Normalization	Original lexicon	Extended lexicon
Books	Mean	0.525	0.567
	Vader, $\alpha = 0.25$	**0.520**	0.586
	Vader, $\alpha = 0.5$	**0.520**	0.575
	Vader, $\alpha = 1$	0.527	0.576
Clothing	Mean	0.504	0.520
	Vader, $\alpha = 0.25$	0.506	0.520
	Vader, $\alpha = 0.5$	**0.501**	0.520
	Vader, $\alpha = 1$	0.505	0.528
Music	Mean	0.536	0.557
	Vader, $\alpha = 0.25$	0.554	0.591
	Vader, $\alpha = 0.5$	0.539	0.571
	Vader, $\alpha = 1$	**0.535**	0.566
Emotional stories	Mean	0.656	0.708
	Vader, $\alpha = 0.25$	**0.607**	0.674
	Vader, $\alpha = 0.5$	0.632	0.692
	Vader, $\alpha = 1$	0.660	0.708

Table 5: Mean Average Error (MAE) on all datasets for original and extended Pattern lexicons using mean and Vader normalization. Lower MAE means better performance. The original Pattern uses a mean function, so that is our baseline. Vader's normalization function improves the performance on all datasets, albeit with different values for hyperparameter α. Extending the lexicon does not improve the performance.

6. Error analysis

Our work started with the assumption that the relatively small lexicon size of Pattern has a big influence on the prediction results; that assumption led to our efforts towards an extension of it. The experiments presented in the previous sections, however, cannot directly confirm if this is the case.

To this end, we selected a subsample of 50 book reviews where the absolute error is more than 3 standard deviations higher than the mean absolute error. On these reviews we conducted an error analysis to identify the reasons for incorrect sentiment prediction. The analysis results, which are shown in Table 6, indicate that the most frequent problem (38% of reviews[14]) is when valenced words from the review text are missing from the lexicon. This validates our hypothesis that increasing the lexicon size should bring a significant performance improvement. Another common issue (30%) we identified was a mismatch between the label (i.e., the review rating) and the text, such as when a very positive review is associated with a negative label. We assume this error is present only in the reviews corpora, since these consist of online user-generated content that we collected automatically from webpages, contrary to the emotional stories corpus, which was collected in a controlled setting.

Other stumbling blocks are those typical of most rule-based systems: the sentiment of a review text is "emergent" or implicit, and the review text does not include specific strongly-valenced words (20%); the lack of word sense disambiguation (18%); rules that do not cover every possible input (e.g. negations of the verb; 12%); multiple and conflicting aspects mentioned in a review, such as when some aspects refer to another item (10%) or to the online service (10%); typographical and

14. Each review can, of course, give rise to multiple types of problems at the same time.

tokenization errors that prevent a match with lexicon entries (4%); and, finally, the usage of irony and sarcasm (4%), whose detection is in itself a challenging research topic (Zhang et al. 2019).

Error type	Number of errors	Percentage
Words from review missing in lexicon	19	38%
Wrong review label	15	30%
No clearly-emotional words in review	10	20%
Pattern uses the wrong word sense	9	18%
Pattern does not recognize negation or intensifier	6	12%
Review discusses (or compares with) different item	5	10%
Review about webshop service (or other meta matter)	5	10%
Typos prevent match with lexicon words	2	4%
Irony	2	4%

Table 6: Error analysis on a subset of 50 book reviews with high MAE. Most errors are due to the limited word coverage

7. Conclusions

Sentiment analysis is an active field of research in Natural Language Processing, but tools for automatically classifying positive and negative texts are routinely used by a larger community of scholars, even outside NLP. Pattern.nl is one of the few off-the-shelf choices for Dutch sentiment analysis, and the only tool that is open source and freely available. However, it was created starting from book reviews, and its performance on different domains might suffer from this (see Boukes et al. (2020) for examples in another domain, i.e. economic news).

In this work, we tried expanding the lexicon by adding the words of Moors et al. (2013). Despite our various attempts, no positive effect of this expansion has been observed, even when taking into account the possible effects of "neutral" words. As previously mentioned, this might mean that our chosen lexicon is not suited for this, but others, such as the one developed by Verheyen et al. (2020), could give better results. In support of this hypothesis is the fact that Vader's lexicon is much larger than many comparable tools, and performs better than these across multiple domains (Hutto and Gilbert 2014). On the other hand, Boukes et al. (2020) claim that a large lexicon of valenced words is not needed per se, as long as "relevant keywords" are in the dictionary. Our initial error analysis indicates that missing lexicon words are a common source of errors in Pattern's predictions, but identifying a minimum list of words to be included in an extended lexicon, so that it performs better across multiple domains, is still a topic for further research.

We also investigated whether there is any benefit in choosing different formulae for aggregating individual word scores and deciding the output score. We found that replacing the original formula, a simple average of all chunks, with the normalization formula used by Vader – and tweaking its hyperparameter α, whose optimal value lies between 0.25 and 0.5 in our experiments, but has proved itself to be highly dataset dependent – improves performance across all datasets and domains. The improvement is minimal for product reviews, but more significant for the corpus of emotional stories. Further work should determine whether other functions for word scores aggregation can improve current results.

Ultimately, we hope that this discussion of positive and negative results can help both researchers and developers of sentiment analysis tools for Dutch and other under-resourced languages.

Acknowledgements

This work is part of the research programmes DATA2GAME (project number 055.16.114) and Affective Language Production (360-89-050), both financed by the Dutch Research Council (NWO). We thank Daphne Rietvelt and Joanna Kruyt for their early explorations and experiments as part of their student projects at the University of Twente, and Nadine Braun at the University of Tilburg for sharing the corpus of emotional stories before publication.

References

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani (2010), SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, *Proceedings of the Conference on International Language Resources and Evaluation (LREC)*, pp. 2200–2204.

Boot, Peter, Hanna Zijlstra, and Rinie Geenen (2017), The Dutch translation of the Linguistic Inquiry and Word Count (LIWC) 2007 dictionary, *Dutch Journal of Applied Linguistics* **6** (1), pp. 65–76, John Benjamins.

Boukes, Mark, Bob van de Velde, Theo Araujo, and Rens Vliegenthart (2020), What's the tone? Easy doesn't do it: Analyzing performance and agreement between off-the-shelf sentiment analysis tools, *Communication Methods and Measures* **14** (2), pp. 83–104, Taylor & Francis.

Bradley, M.M. and P.J. Lang (1999), Affective norms for English words (ANEW): Instruction manual and affective ratings, *tech. report C-1*, University of Florida.

Braun, Nadine, Martijn Goudbeek, and Emiel Krahmer (2020), Own- and other-annotation of emotions in text. https://osf.io/ekqmj.

De Smedt, Tom and Walter Daelemans (2012a), Pattern for Python, *The Journal of Machine Learning Research* **13** (1), pp. 2063–2067, JMLR.org.

De Smedt, Tom and Walter Daelemans (2012b), "Vreselijk mooi!" (terribly beautiful): A subjectivity lexicon for Dutch adjectives, *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, pp. 3568–3572.

Delobelle, Pieter, Thomas Winters, and Bettina Berendt (2020), RobBERT: A Dutch RoBERTa-based language model, *arXiv:2001.06286 [cs.CL]*. https://arxiv.org/abs/2001.06286.

Hogenboom, Alexander, Bas Heerschop, Flavius Frasincar, Uzay Kaymak, and Franciska de Jong (2014), Multi-lingual support for lexicon-based sentiment analysis guided by semantics, *Decision Support Systems* **62**, pp. 43 – 53. http://www.sciencedirect.com/science/article/pii/S0167923614000645.

Hutto, Clayton J and Eric Gilbert (2014), Vader: A parsimonious rule-based model for sentiment analysis of social media text, *Proceedings of the 8th International AAAI Conference on Weblogs and Social Media (ICWSM-14)*, Oxford, United Kingdom, pp. 216–225.

Moors, Agnes, Jan De Houwer, Dirk Hermans, Sabine Wanmaker, Kevin van Schie, Anne-Laura Van Harmelen, Maarten De Schryver, Jeffrey De Winne, and Marc Brysbaert (2013), Norms of valence, arousal, dominance, and age of acquisition for 4300 Dutch words, *Behavior Research Methods* **45** (1), pp. 169–177. http://dx.doi.org/10.3758/s13428-012-0243-8.

Neviarouskaya, Alena, Helmut Prendinger, and Mitsuru Ishizuka (2011), Affect analysis model: Novel rule-based approach to affect sensing from text, *Natural Language Engineering* **17** (1), pp. 95–135, Cambridge University Press.

Pennebaker, J. and M. Francis (2001), Linguistic inquiry and word count: LIWC. Erlbaum Publishers.

Pontiki, Maria, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit (2016), SemEval-2016 task 5: Aspect based sentiment analysis, *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California, pp. 19–30.

Schrauwen, Sarah (2010), Machine learning approaches to sentiment analysis using the Dutch Netlog Corpus, *Technical report*, Computational Linguistics and Psycholinguistics Research Center.

Stone, P.J., D.C. Dunphy, and M.S. Smith (1966), *The General Inquirer: A Computer Approach to Content Analysis*, MIT press.

Thelwall, Mike, Kevan Buckley, and Georgios Paltoglou (2012), Sentiment strength detection for the social web, *Journal of the American Society for Information Science and Technology* **63** (1), pp. 163–173, Wiley Online Library.

Van Atteveldt, Wouter, Jan Kleinnijenhuis, Nel Ruigrok, and Stefan Schlobach (2008), Good news or bad news? Conducting sentiment analysis on Dutch text to distinguish between positive and negative relations, *Journal of Information Technology & Politics* **5** (1), pp. 73–94, Taylor & Francis.

Van der Burgh, Benjamin and Suzan Verberne (2019), The merits of universal language model fine-tuning for small datasets – a case with Dutch book reviews, *arXiv:1910.00896 [cs.IR]*. https://arxiv.org/abs/1910.00896.

Verheyen, Steven, Simon De Deyne, Sarah Linsen, and Gert Storms (2020), Lexicosemantic, affective, and distributional norms for 1,000 Dutch adjectives, *Behavior Research Methods* **52** (3), pp. 1108–1121. https://doi.org/10.3758/s13428-019-01303-4.

Vries, Wietse de, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim (2019), BERTje: A Dutch BERT Model, *arXiv:1912.09582 [cs.CL]*. http://arxiv.org/abs/1912.09582.

Zhang, Lei, Shuai Wang, and Bing Liu (2018), Deep learning for sentiment analysis: A survey, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **8** (4), pp. e1253, Wiley Online Library.

Zhang, Shiwei, Xiuzhen Zhang, Jeffrey Chan, and Paolo Rosso (2019), Irony detection via sentiment-based transfer learning, *Information Processing & Management* **56** (5), pp. 1633–1644, Elsevier.